

**IN THE DRAWINGS:**

**Formal drawings have been previously been supplied for this application.**

REMARKS

In response to the Examiner's Office Action of August 30, 2004, Applicant is herein presenting his considerations and response to the Examiner's comments.

The Examiner has indicated that claims 1-7, 10-20, 23-26, 28-34, and 37-40 are rejected under 35 USC 103(a) as unpatentable over Butts (W097/37303) in view of Matthews (U.S. Patent 5,974,256).

Applicant would now again traverse the Examiner's conclusions on the above claims.

Further, Examiner has rejected claim 27 under 35 USC 103(a) as unpatentable over Matthews, U.S. Patent 5,974,256 in view of so-called "admitted prior art" (APA). Here again, in view of the amendments to the claims, Applicant would again traverse the Examiner's conclusions on this claim.

In the Applicant's prior amendment dated June 14, 2004, Applicant gave considerable address to the Butts reference and the Matthews reference, and the inapplicability of these references to the present invention, and also to the impropriety of combining Butts and Matthews.

It is the consideration of Applicant that the Examiner is improperly combining the Butts and the Matthews references and moreover, is not reading the broadest claims (claims 1, 14, 27 and 28), in the most comprehensive fashion.

As a result, Applicant has provided clarifying claim amendments to the independent claims 1, 14, 27 and 28 in order that the Examiner may take a more comprehensive look in regard to the claimed invention.

At paragraph 5 of the Examiner's Office Action, the Examiner indicates that Butts does not explicitly teach creating translatable source code, utilizing the translatable source code to produce a series of software components while Matthews teaches creating a translatable source code (JAVA code snippet at column 5, lines 13-25) using the translatable source code to produce a series of software components (JAVA source files; column 5, lines 37-48).

Here, the Examiner contends that Matthews, U.S. Patent 5,974,256, discloses the feature of creating a translatable source code and utilizing the translatable source code to produce a series of software components.

However, as was indicated in Applicant's response to the prior Office Action, it should be noted that the disclosure in Matthews makes no mention of creating an executable component from legacy source code.

The Matthews reference discloses a method for translating a Graphical User Interface resource data file into JAVA code. But note --- the invention disclosed in Matthews only teaches a translation tool which migrates or maps Graphical User Interface resource information from a Windows™ based resource file to JAVA native code to facilitate the migration of existing native applications.

In other words, the invention described in Matthews only translates sections of code which are explicitly related to the Graphical User Interface Resource files of a particular application from one specific

environment (i.e., Microsoft Windows™) to an operating system independent language, namely JAVA.

The Graphical User Interface resource files in a Windows application only represents a small portion of the total code required to create a Windows Graphical User Interface application.

This is admitted in the Matthews reference at column 9 line 63, to column 10 line 4, where it is stated, in part --- "a set of 'word code' must still be incorporated in order to provide a working interface".

The Matthews reference does not suggest or teach towards the desirability of creating an executable application from legacy source code. Matthews is wholly concerned with the system that maps Windows-based Graphical User Interface resource files to JAVA code.

There is no teaching, implicitly or explicitly, that such a tool may be used to create an executable application that can interface with legacy applications. Matthews teaches an invention that is merely the first step in migrating a Graphical User Interface from a Windows-based environment to a JAVA environment.

Applicant has now slightly amended the independent claims, namely claims 1, 14, 27 and 28, which now clearly state the feature of "utilizing said legacy source code to produce a series of executable software components that provide the functionality for interaction with the legacy software application . . .".

As was indicated above, this particular feature is certainly not disclosed in Matthews, and is certainly not disclosed even if the Matthews' reference is combined with the Butts reference. Therefore, it should be emphasized that Applicant's independent claims, as a whole, are novel and non-obvious, notwithstanding any combination on the part of the Examiner to combine the Matthews and the Butts references. And as was indicated in Applicant's previous amendment, it is improper to combine these references barring any suggestion that these references should be combined with each other.

With the above commentary and differentiations in mind, it is requested that the Examiner now consider Applicant's invention as a whole in its entirety, and subsequently provide a timely Notice of Allowance for the extant claims.

Respectfully submitted,

By Alfred W. Kozak  
Alfred W. Kozak  
Reg. No. 24,265  
(858) 451-4615  
(949) 380-5822

---

Certificate of Mailing (37 CFR 1.8a)

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: MAIL STOP AMENDMENT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Date: November 29, 2004

Patti S. Preddy  
Patti S. Preddy

CLEAN VERSION OF AMENDED CLAIMS:

Amended independent claims:

1. A method for adapting a legacy software application, created from legacy source code and developed for an environment comprising a centralized computing resource interconnected to a series of computer terminal devices, to a network environment, wherein said network environment comprises a system of distributed, interconnected network computing resources, said method comprising the steps of:

utilizing said legacy source code to produce a series of executable software components that provide the functionality for interaction with the legacy software application, said components being executable by at least one of said computing resources in said network environment, and wherein upon execution, said computing resource is caused to interconnect with said legacy software application over said network so as to interact with said legacy software application in the transmission or receipt of information to and from said legacy software application.

14. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for adapting a legacy software application created from legacy source code and developed for an environment comprising a centralized computing resource interconnected to a series of computer terminal devices, to a network environment, wherein said network environment comprises a system of distributed, interconnected network computing resources, said method comprising the steps of:

utilizing said source code to produce a series of executable software components that provide the functionality for interaction with the legacy software application, said components being executable by at least one of said computing resources in said network environment, and wherein upon execution, said computing resource is caused to interconnect with said legacy software application over said network so as to interact with said legacy software application in the transmission of receipt of information to and from said legacy software application.

27. A method for adapting a 4GL legacy software application including template definitions from which a legacy software application can be generated, comprising the steps of;

utilizing said template definitions to produce a series of software executable components, said components being executable by at least a computing resource in a network environment comprising a system of distributed, interconnected network computing resources, and wherein upon execution, said computing resource is caused to interconnect with the 4GL legacy software applications so as to interact with the legacy application in the transmission and receipt of information to and from the legacy application.

28. A system for adapting a legacy software application, created from legacy source code and developed for an environment comprising a centralized computing resource interconnected to a series of computer terminal devices, to a network environment, wherein said network environment comprises a system of distributed, interconnected network computing resources, the system comprising;

means utilizing said legacy source code to produce a series of executable software components for providing the functionality for interaction with the legacy software application, said components being executable by at least one of said computing resources in said network environment, and wherein upon execution, said computing resource is caused to interconnect with said legacy software application over said network so as to interact with said legacy software application in the transmission or receipt of information to and from said legacy software application.